(19) Europäisches Patentamt
European Patent Office
Office européen d s brevets

(11) EP 0 913 757 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**06.05.1999 Bulletin 1999/18**

(51) Int Cl.6: **G06F 1/00**

(21) Application number: **98308688.5**

(22) Date of filing: **23.10.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **30.10.1997 US 960867**

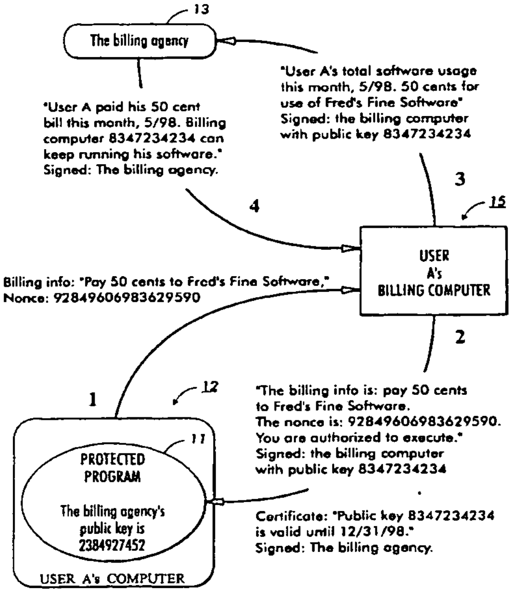
(71) Applicant: **XEROX CORPORATION**
**Rochester, New York 14644 (US)**

(72) Inventor: **Merkle, Ralph C.**
**Sunnyvale, CA 94087 (US)**

(74) Representative: **Skone James, Robert Edmund**
**GILL JENNINGS & EVERY**
**Broadgate House**
**7 Eldon Street**
**London EC2M 7LH (GB)**

(54) **Protected software**

(57) The functionality of "protected shareware" is controlled by digitally signed messaging protocol. Protective code within the shareware controls the functionality of the shareware in response to authorization messages which are supplied directly or indirectly by the shareware supplier. These messages are digitally signed in whole or part by or on behalf of the shareware supplier using the supplier's secret signing key. The shareware, in turn, includes the public checking key for this digital signature of the supplier, thereby enabling the protective code to authenticate any such authorization message before acting in reliance upon it. The shareware includes an integrity self-checking routine which is run at appropriate times to ensure that shareware, including its protective code, is in an anticipated state. Typically this is equivalent to a finding that the code has not been altered, but it is conceivable that the code might be dynamically altered in some authorized (i.e., "anticipated") way.



"User A's total software usage this month, 5/98. 50 cents for use of Fred's Fine Software"
Signed: the billing computer with public key 8347234234

"User A paid his 50 cent bill this month, 5/98. Billing computer 8347234234 can keep running his software."
Signed: The billing agency.

Billing info: "Pay 50 cents to Fred's Fine Software,"
Nonce: 92849606983629590

"The billing info is: pay 50 cents to Fred's Fine Software.
The nonce is: 92849606983629590.
You are authorized to execute."
Signed: the billing computer with public key 8347234234

Certificate: "Public key 8347234234 is valid until 12/31/98."
Signed: The billing agency.

*FIG. 1*

EP 0 913 757 A2

Printed by Jouve, 75001 PARIS (FR)

## Description

[0001] This invention relates to countermeasures for combating computer software "piracy" and, more particularly, to techniques for protecting executable computer programs (including informally distributed programs) against unauthorized use.

[0002] The distribution of computer software fundamentally involves two parties: the software author or the author's successor in interest (i.e., the "supplier") and a prospective end user. In a standard commercial transaction, the end user usually is motivated to acquire a legitimate right to use identified software in a particular computing environment on a satisfactory schedule, while the supplier generally is interested in receiving an acceptable return on the transaction. Various formalized distribution strategies, such as the use of retail outlets, direct sales, mail order distribution, etc., have been developed to resolve the competing interests of these parties, but many of the more popular distribution techniques tend to burden the distribution process with costs which add little, if any, value to the underlying software. For example, software suppliers frequently incur packaging and handling costs which add no substantive value to the software, but which add to the cost of delivering the software to the end users.

[0003] Furthermore, conventional commercial software distribution strategies too often fail to provide adequate protection against unauthorized use of the software. A few suppliers still "copy protect" their program offerings, but this practice has fallen into general disfavor because it restricts the ability of legitimate users to configure and backup their computing systems in accordance with their individual needs and preferences. Thus, some software suppliers have elected to forego the use of built in anti-piracy mechanisms all together, while others have adopted inconvenient countermeasures that are somewhat less objectionable to the user community. For example, some programs are designed to run only if the host computer has a program specific "dongle" on a designated port, other programs are designed to run only if the user correctly responds to a query which tends to verify the legitimacy of the user (such as a query requesting information from a specific page of the program documentation), and still other programs are designed to run only after they have been "personalized" with, say, a user name and/or address.

[0004] Computer "shareware" is distributed more informally and with less overhead cost than standard commercial software. The user-to-user distribution process that shareware tends to promote is at least arguably better aligned with the fundamental character of executable computer programs than the direct marketing/retail distribution model that now dominates the distribution of commercial software because an executable program essentially is intangible information which is represented by a pattern of binary "1's" and "0's". Others have recognized the potential of shareware as a vehicle for distributing commercial software, but it has been difficult to capitalize in the commercial market on the efficiency and effectivity of this distribution model because there has been no easily implemented, built-in protection mechanism for enforcing the software suppliers' compensation expectations.

[0005] Mori et al., "Superdistribution: The Concept and Architecture", The Transactions of the IEICE, Vol. E73, No. 7, July 1990, pp. 1133-1146 addresses this enforcement issue, but it suggests encrypting the software. Unfortunately, however, when an ordinary key encryption technique is employed for encrypting software which is subject to unrestricted distribution for use by any user, then the key for decrypting the software must necessarily also be available to any user. Mori et al., propose the use of stringent physical security measures for reducing the risk of divulging key contents to the users, but system security is questionable at best if a single common key must be available to all host computers on which the encrypted shareware might run. If, on the other hand, multiple keys are used, then multiple versions of the software must be created and distributed which is often a severe inconvenience. Also see, Mori et al., "Superdistribution: An Electronic Infrastructure for the Economy of the Future", Trans. of Information Proc. Soc. of Japan, Vol. 38, No. 7, July 1997, pp 1465 - 1472.

[0006] Consequently, there still is a need for a cost effective response to the challenge of affording computer software suppliers adequate protection against unauthorized use of their executable computer program offerings when unrestricted distribution and installation of these programs is permitted (i.e., when a shareware distribution model is employed). For commercial applications, it would be desirable to have a system solution for automatically carrying out at least the basic elements of a commercial transaction, including communicating the supplier's asking price for program usage rights to the prospective user, registering the user's payment or promise to pay, and denying or revoking the user's usage rights if payment is not timely received. Furthermore, it would be advantageous for this solution to perform these functions even if the system resources which enforce the supplier's interests only have intermittent access to the host computer on which the user decides to run the shareware program.

[0007] In accordance with the present invention, a method for protecting an executable computer program from unauthorized use independently of any methodology for distributing said program to prospective users comprises the steps of: inhibiting at least some functional feature of said program from running on a compatible stored program computer until said computer receives a message that is digitally signed by an authorized party using a secret signing key; providing said program access to a public checking key for said secret signing key in said program; running an integrity self-check over said program to confirm that said program is in an anticipated state; communicating a message that is digitally signed with said secret signing key to said computer for tentatively approving the running of said functional feature of said program on said computer; applying said public checking key to said digitally signed message for authenticating it; and enabling said functional feature to run on said selected computer in response to the authentication of said message, provided that it has been confirmed that said program is in said anticipated state.

**[0008]** Thus, for example, the functionality of "protected shareware" is controlled by digitally signed messaging protocol. Protective code within the shareware controls the functionality of the shareware in response to authorization messages which are supplied directly or indirectly by the shareware supplier. These messages are digitally signed in whole or part by or on behalf of the shareware supplier using the supplier's secret signing key. The shareware, in turn, includes the public checking key for this digital signature of the supplier, thereby enabling the protective code to authenticate any such authorization message before acting in reliance upon it. The shareware includes an integrity self-checking routine which is run at appropriate times to ensure that shareware, including its protective code, is in an anticipated state. Typically this is equivalent to a finding that the code has not been altered, but it is conceivable that the code might be dynamically altered in some authorized (i.e., "anticipated") way.

**[0009]** Additional objects and advantages of this invention will become apparent when the following detailed description is read in conjunction with the attached drawings, in which:

Figure 1 diagrammatically illustrates a process for applying the present invention to commercial shareware;

Figure 2 schematically illustrates a hardware implementation of the process shown in Figure 1; and,

Figures 3 to 5, are simplified logical flow diagrams for a protected program, a billing agency and a billing computer, respectively, which communicate (as indicated by the dashed lines) to carry out a version of this invention.

**[0010]** Turning now to the drawings, and at this point especially to Figure 1, a somewhat elaborated embodiment of the invention is shown to illustrate a suitable implementation for simple commercial applications. However, at a more fundamental level, it will be observed that an executable computer program 11 that is protected in whole or part in accordance with this invention includes embedded protective code which interacts with the existing state of the runtime environment for the program 11 in accordance with predetermined rules. These rules, in turn, are selected to cause the protective code to modify the functionality of the program 11 in predetermined ways as a function of predefined, recognizable changes in the state of the runtime environment.

**[0011]** For example, in a relatively straightforward application of this invention, the protective code might simply limit the functionality of the protected program 11 unless and until an authorization message is received, directly or indirectly, from a source 13 which is trusted by the program supplier, such as a supplier controlled computer. In practice, the communication channel between the protected program 11 and the trusted source 13 may not be secure, so the source 13 suitably authenticates each of the authorization messages that it issues on behalf of the program supplier by digitally signing each of them with the program supplier's secret signing key. Then, to enable the protective code for the program 11 to verify the authenticity of any apparent authorization message that it receives prior to acting in reliance on it, the protected program 11 includes the public checking key for the supplier's secret signing key. This public checking key may be authenticated by the integrity self-check. Alternatively, the program 11 may include a hash value for this public checking key and have provision for authenticating the public checking key from this hash value. Moreover, still other known authentication methods can be used to authenticate the supplier's public checking key.

**[0012]** As will be appreciated, the interaction between the protective code and the runtime environment for the protected program 11 need not be a simple binary function. Instead, the protective code may give the user access to incrementally differing functional performance capabilities of the program 11 in response to different prespecified states of the runtime environment. Furthermore, the protective code may be programmed to modify its response to any one or more of the recognizable states of the runtime environment for the program 11 as a function of time and/or as a function of any other appropriately monitored variable. For instance, the protective code may be programmed to engage the source 13 in a re-authorization process under predetermined conditions, such as after the protected program 11 has been used pursuant to a prior authorization for a fixed period of real time or run time, or upon the arrival of a date certain.

**[0013]** In keeping with one of the important features of this invention, provision is made for reducing the risk of the protective code for the program 11 from being bypassed or overridden. To this end, the protected program 11 advantageously includes an integrity self-checking mechanism which invokes predetermined countermeasures (e.g., functionally inhibiting the protected program 11 and/or launching warning messages to the program supplier or the supplier's agent, such as the trusted source 13) if it is found that the integrity of the program 11 has been compromised. For example, the protected program 11 might include a check sum routine for computing a sum over the existing version of the program code at more or less randomly selected times and for then comparing each of these computed sums against a precomputed integrity reference so that appropriate countermeasures can be taken if there is any discrepancy. While this integrity self-checking may not provide fail-safe protection for the protective code, the self-checking may be sufficiently sophisticated to create a meaningful economic barrier against code altering attacks. For instance, a series of integrity checks can be implemented so that every integrity check, except for the last, is monitored by the next integrity check in this series. As will be understood, the code for implementing such an integrity check can be embedded virtually anywhere within the protected program 11 as a distributed or non-distributed routine.

**[0014]** The issuance of a properly authenticated authorization message by the source 13 may be preconditioned upon the source 13 being directly or indirectly notified that there is at least apparent compliance with certain prerequisites. Accordingly this invention can be employed for protecting executable computer programs against unauthorized use whenever the suppliers of such programs are prepared to grant program usage rights subject to apparent compliance with predetermined, monitorable conditions. It may not be practical or even feasible to provide absolute compliance guarantees, but compliance assurances of sufficient reliability to reasonably protect the program suppliers' interests usually can be provided. Therefore, this invention may result in greater reliance on the shareware software distribution model, including by program suppliers who are specifically interested in being paid for the use of their programs.

**[0015]** In keeping with the commercial flavor of the illustrated embodiment, the source 13 of the digitally signed authorization messages is shown as being a central billing agency which represents the interests of the supplier of the protected program 11, possibly together with the interests of the suppliers of other similarly protected programs (not shown). Prospective users of the protected program 11 could be required to contact this billing agency 13 for authorization prior to being given access to the protected features of the program 11. However, such a pre-approval process might be found to be too restrictive and could even prove to be counterproductive because a substantial portion of the prospective users may not have convenient, more or less real time access to the billing agency 13. Thus, an authorization process which requires only occasional interaction with the central billing agency 13 is favored. For example, a process that involves regularly scheduled contact, such as monthly contact, with the billing agency 13 may strike a reasonable balance between the avoidance of user inconvenience and the protection of the program supplier's interests.

**[0016]** To give the program supplier some flexibility in establishing and revising the terms and conditions that govern an authorized use of a protected program, such as the program 11, provision is made for storing a selected set of accounting routines in each prospective user's local computing environment (i.e., in the user's host computer 12 or in a peripheral unit, as at 15 in Figure 2, that is interfaced with the computer 12 as more fully described hereinbelow). The protected program 11, in turn, includes calls to preselected ones of these routines for communicating the supplier's terms and conditions for authorized use of the program 11 to the prospective user, such as via the standard display interface of the computer 12, and for recording the user's acceptance and/or rejection of this offer in a suitable memory. For example, the supplier might specify that the protected program can be used for 30 days at no charge and establish a $50 annual or paid-up license fee for any use beyond this initial free trial period. As will be understood, a clock, such as shown at 16 in Figure 2, is a useful tool to have for tracking the time periods specified by such offers, but the supplier's expectations and the user's acceptance of such an offer otherwise are self-defined. Thus, additional calls to the accounting routines are included in the protected program 11 for causing data pertaining to any usage of the protected program 11 to be recorded in tamper resistant, non-volatile memory, such as the non-volatile random access memory (RAM) 17 in Figure 2. This usage data, in turn, is communicated to the central billing agency 13 on a selected schedule via a suitable communication link. For example, a modem 18 may be employed for communicating the usage data to the billing agency via the public switched telephone network.

**[0017]** Alternatively, the accounting and billing functions could be performed internally to the user's computer 12. This could be done by supplementing an ordinary personal computer or workstation with a secure kernel that operates with protected memory which is inaccessible to the user except through the approved accounting routines. It is believed that programmers who are generally familiar with the concepts underlying such software could readily implement a secure process for handling the limited task of billing information in software. This, of course, would reduce the cost and complexity of any special purpose hardware that might be required to carry out this invention.

**[0018]** As illustrated, however, the computer 12 (e.g., a personal computer, a computer workstation, or any other type of stored program controlled computer) of each prospective user of the protected program 11 has a serial or parallel data port which is interfaced with a special purpose billing computer 15. As shown in further detail in Figure 2, this billing computer 15 has a processor 21 for managing all memory transactions involving the non-volatile RAM 17, a read-only memory (ROM) 22, and some additional volatile or non-volatile RAM 23. The non-volatile RAM 17 stores the usage data, and the ROM 22 stores the accounting routines. Thus, to prevent unauthorized manipulation of this information, and of any "scratch pad" data that may be written into the RAM 23, the billing computer 15 advantageously is designed to be "tamper resistant", using any of the known techniques for providing such tamper resistance. The clock 16 then is desirably included in the billing computer 15 to take advantage of this tamper resistance. The modem 18, on the other hand, may be internal of or external to the billing computer 15. If an internal modem is employed, it suitably is plugged into a standard telephone interface (not shown) and configured to allow telephone signals to pass to and from other equipment (also not shown) when it is in an idle state. In practice, it may be most cost effective to use embedded processor technology to implement the billing computer 15 as a "single chip" computer.

**[0019]** Conceivably, an attack against the billing computer 15 could result in the divulgation of its secret signing key. Even then, however, the protected program 11 could protect itself against a fraudulent manipulation of the date as known to the billing computer 15 by checking the expiration date on the authentication certificates that it receives from the billing agency 13 via the billing computer 15 against a date inferred from its operating environment, such as by checking the dates on the most recent data files. The use of these authenticating certificates is describe in some additional detail hereinbelow.

**[0020]** Returning to Figure 1, communications between the billing agency 13 and the billing computer 15 generally are carried by unsecure communication links. Thus, to ensure the integrity of the information they exchange, the billing agency 13 and the billing computer 15 typically not only digitally sign the messages they transmit to each other with their respective secret signing keys, but also have a priori knowledge of each others public checking keys so that they can validate those messages upon receipt.

**[0021]** If desired, a software program (not shown) can be provided to enable the user to interrogate the billing computer 15 for data relating to the current or accumulated usage charges that the user has incurred, the payment status of the user's account, etc. This program may be a stand alone program or a utility that is included in the protected program 11. The user is merely given tamper resistant, read only access to the billing data, so the billing interrogation program need not be secure.

**[0022]** The authenticity of the message traffic between the billing agency 13 and the billing computer 15 is assured by the above-described digital signature protocol. However, there still is a risk of a successful "spoofing" attack against the protected program 11, unless provision is made for authenticating any message which purports to authorize the user's computer 12 to execute any of the protected code of the program 11. For example, a fraudulent authorization message might be communicated to the computer 12 by an imitator (not shown) of the billing computer 15 whose behavior would be unpredictable at best. As previously pointed out, the public checking key for the digital signature of the billing agency 13 (i.e., the supplier's digital signature) is easily embedded in all unaltered copies of the protected program 11. In this embodiment, however, the protected program 11 indirectly communicates with the central billing agency 13 via the billing computer 15. Therefore, to enable the protected program 11 to use the billing agency's public checking key for validating authorization messages which appear to the host computer 12 to be from the billing computer 15, one or more of the messages that the billing computer 15 communicates to the host computer 12 includes a certificate or the like which is digitally signed by the billing agency 13. For example, as shown, this certificate passes the public checking key for the billing computer's secret signing key to the protected program 11 while certifying the validity of the public checking key for a limited time. Thus, the program 11 can use this public checking key to authenticate the digitally signed messages that it receives from the billing computer 15, unless and until the billing agency's certification of the billing computer 15 expires.

**[0023]** As a general rule, it is needlessly inefficient for the billing agency 13 to actively participate in preparing each and every authorization message that is transmitted to the host computer 12 by the billing computer 15. For this reason, the billing agency 13 may serially issue term limited authenticating certificates to the billing computer 15 in accordance with a predetermined schedule that is selected to ensure that an unexpired certificate is always in force, unless the billing agency 13 or the program supplier has decided to terminate the user's authorization to use the protected features of the program 11, such as for non-payment of the usage fees. To accommodate this methodology, the billing agency 15 stores the most recently received authenticating certificate for use as needed. Then, when an authorization message is received by the host computer 12, the protected program 11 first checks the expiration date of the authenticating certificate for received message against the current date according to the clock 16 to confirm that the certificate is unexpired prior to taking any further action in reliance upon the apparent authorization message. Advantageously, the protected program 11 includes code for causing the host computer 12 to contact the billing computer 15 for an authorization message each time the program 11 is initialized and/or periodically while the program 11 is running, thereby ensuring a more or less timely reassertion of the protection for the program 11 if the user's usage rights are revoked or expire.

**[0024]** Communications from the protected program 11 to the billing computer 15 generally are unsecure. For this reason, the billing computer 15 advantageously echoes and digitally signs all messages that it receives from the host computer 12 for the program 11. This enables the program 11 and the computer 12 to self-authenticate messages the billing computer 15 believes were received from them because the echoed messages are returned to the host 12 via an authenticated channel (i.e., in messages that are digitally signed by the billing computer 15).

**[0025]** For thwarting "replay" attacks against the protection for the software 11, random information, such as a so-called "nonce", is beneficially included in each message that is sent to the billing computer 15 under the control of the protected software 11. This nonce suitably is supplied by the software 11 through the use of a random number generating routine. The billing computer 15, in turn, echos this random information for return to the program 11 as described above.

**[0026]** In the illustrated embodiment, credit is extended to the user because the usage fees are computed by the billing agency 13 on the basis of the usage data that is accumulated by the billing computer 15. Thus, an unscrupulous user could avoid paying usage fees by destroying the billing computer 15 before the usage data is communicated to the billing agency 13. For this reason, the billing computer 15 might be programmed to enforce a credit limit on the user.

**[0027]** In operation, the billing agency 13 notifies the billing computer 15 in a digitally signed message when payments on the user's account are received. This enables the billing computer 15 to maintain a running credit balance on the user's account. Consequently, any balance due on the user's account can be aged by the billing computer 15 through the use of the clock 16. Thus, delinquency rules may be programmed into the billing computer 15 for determining whether future requests for authorization to use the protected program 11 on the host computer 12 should be honored or denied. As will be recalled, denying a re-authorization request effectively terminates the host computer's ability to execute the protected features of program 11 and, therefore, is a useful mechanism for enforcing the program supplier's rights.

**[0028]** If desired, the protected program 11 can include code for negotiating the terms and conditions for use of the protected features of the program 11 with prospective users. For example, the protected program 11 might include code for presenting a prospective user with one after another of a plurality of feature and price differentiated sets of terms and conditions for usage of the program 11, thereby giving the prospective user the freedom to "accept" any one or none of these "offers". Or, the code for these preliminary interactions with prospective users might provide a menu of individually priced, user selectable features of the program 11. The accounting routines and the authorization protocols that are used by the billing computer 15 to manage these negotiated transactions would be designed to provide the differentiation that is necessary to discriminate between the different options the users may select, but otherwise these negotiated transactions are governed by the same general principles as the simple commercial applications that are described above.

**[0029]** An embodiment of the invention is illustrated by the simplified logical flow diagrams that are depicted by Figures 3 to 5 (the messages that are exchanged by these processes are depicted in dashed lines). As will be appreciated, the billing agency 13 is a trusted surrogate for the shareware supplier and, therefore, can be viewed as being one in the same with shareware supplier, unless the billing agency 13 is servicing more than one shareware supplier. The billing computer 15, on the other hand, is a tamper resistant agent of the shareware supplier, so it is subservient to the billing agency 13 (i. e., the billing agency is the superior authority). Alternative embodiments that are consistent with this general framework will readily suggest themselves. For example, significantly more authority for managing the reaction of the protected shareware 11 to the behavior of the user is delegated to the billing computer 15 in the embodiment shown in Figures 1 and 2 than in the embodiment shown in Figure 3. Different designers may have differing views on the desirability of centralizing or delegating various aspects of this management control. If there is a generally agreed upon division of responsibilities between the billing agency 13 and the billing computer 15, then the same billing computer 15 and the same billing agency 15 can be employed to service multiple shareware offerings of the same or different suppliers by simply extending the same general support framework to each of those offerings.

**[0030]** In view of the foregoing, it will be understood that this invention effectively provides a low cost software distribution technique which has many of the most desirable characteristics of the existing shareware distribution model, but which has the advantage of more strongly enforcing the program supplier's rights, such as the right to be paid usage fees in commercial applications.

## Claims

1. A method for protecting an executable computer program from unauthorized use independently of any methodology for distributing said program to prospective users; said method comprising the steps of:

   inhibiting at least some functional feature of said program from running on a compatible stored program computer until said computer receives a message that is digitally signed by an authorized party using a secret signing key;
   providing said program access to a public checking key for said secret signing key in said program;
   running an integrity self-check over said program to confirm that said program is in an anticipated state;
   communicating a message that is digitally signed with said secret signing key to said computer for tentatively approving the running of said functional feature of said program on said computer;
   applying said public checking key to said digitally signed message for authenticating it; and
   enabling said functional feature to run on said

selected computer in response to the authentication of said message, provided that it has been confirmed that said program is in said anticipated state.

2. The method of claim 1, wherein:

said message is supplied by an agent and is digitally signed by said agent using a first secret signing key;
said agent's authority to issue said message is certified by a superior authority for by a certification that is signed with a second secret signing key and which authenticates a public checking key for said first signing key;
said certification is communicated to said program;
said program has access to and is able to authenticate a public checking key for said second secret signing key, thereby enabling said program to authenticate said certification.

3. The method of claim 2, wherein:
said public checking key for said second secret signing key is embedded in said program, whereby said integrity self-check authenticates said public checking key for said second secret signing key.

4. The method of claim 2 or claim 3, wherein:
said agent participates in a commercial transaction with each user to offer the user an opportunity to run said functional feature of said program in return for certain consideration and to monitor the user's return of said consideration if said offer is accepted.

5. The method of claim 4 wherein said commercial transaction involves a multiple option offer which covers multiple functional features of said program and multiple pricing plans.

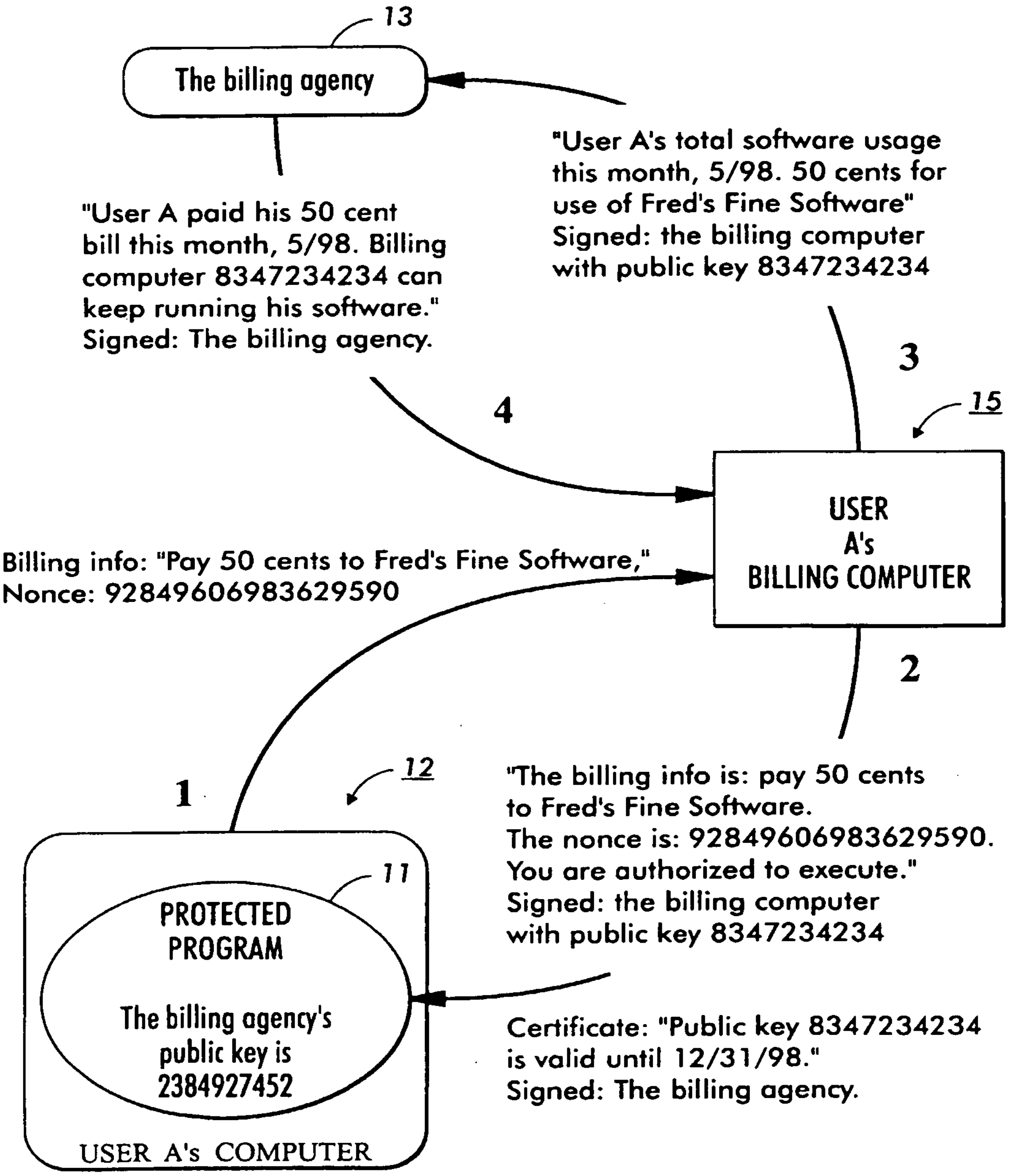

13
The billing agency
"User A's total software usage this month, 5/98. 50 cents for use of Fred's Fine Software" Signed: the billing computer with public key 8347234234
"User A paid his 50 cent bill this month, 5/98. Billing computer 8347234234 can keep running his software." Signed: The billing agency.
3
4
15
USER A's BILLING COMPUTER
Billing info: "Pay 50 cents to Fred's Fine Software," Nonce: 92849606983629590
2
1
12
"The billing info is: pay 50 cents to Fred's Fine Software. The nonce is: 92849606983629590. You are authorized to execute." Signed: the billing computer with public key 8347234234
11
PROTECTED PROGRAM
The billing agency's public key is 2384927452
Certificate: "Public key 8347234234 is valid until 12/31/98." Signed: The billing agency.
USER A's COMPUTER

FIG. 1

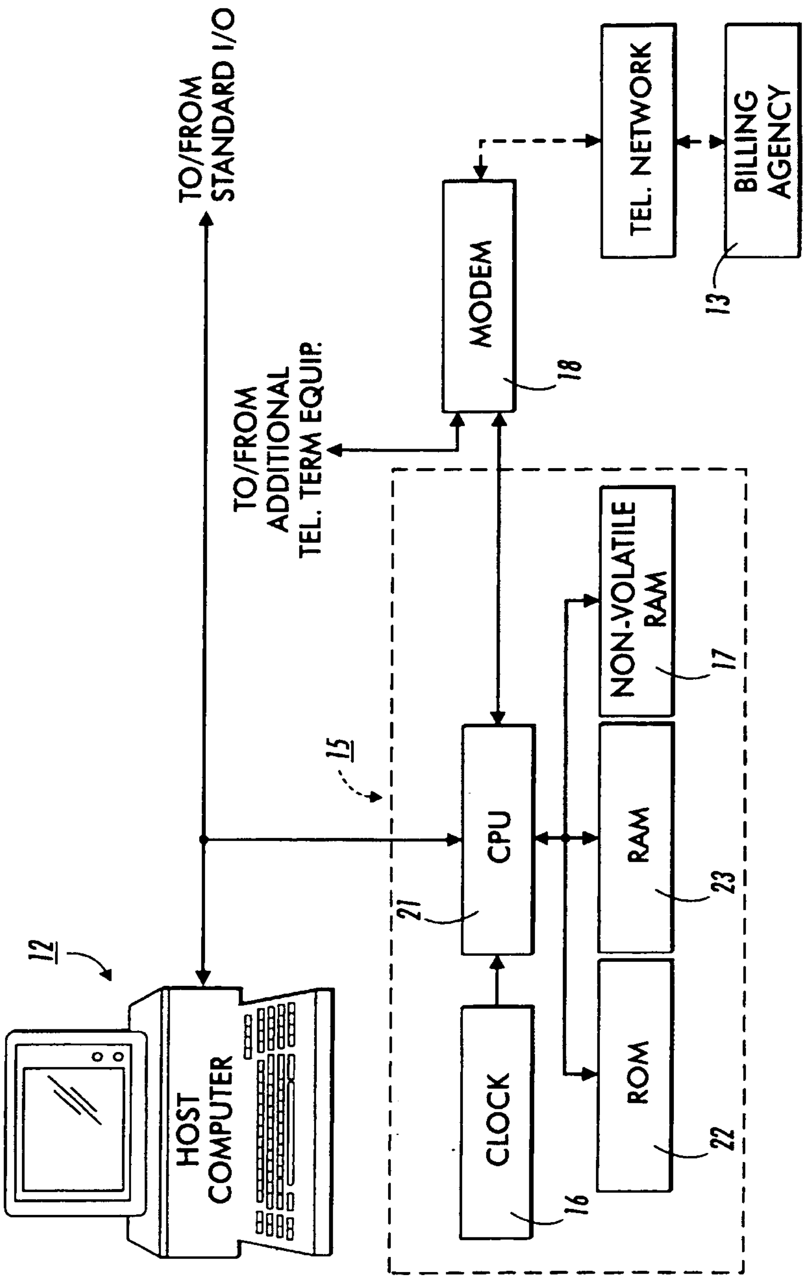

TO/FROM
STANDARD I/O
TO/FROM
ADDITIONAL
TEL. TERM EQUIP.
MODEM
18
TEL. NETWORK
BILLING
AGENCY
13
12
HOST
COMPUTER
15
21
CPU
CLOCK
16
ROM
22
RAM
23
NON-VOLATILE
RAM
17

FIG. 2

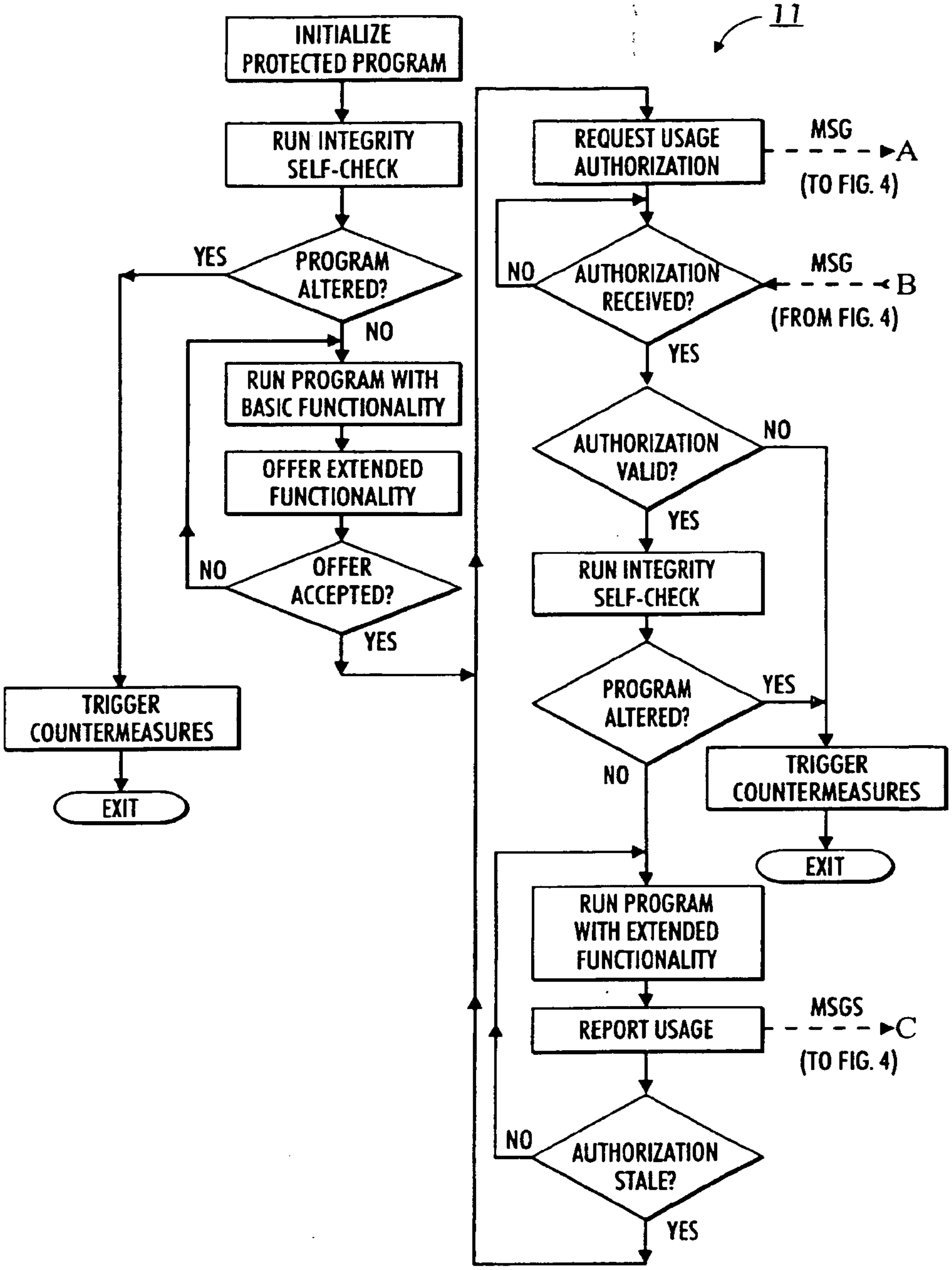

11
INITIALIZE PROTECTED PROGRAM
RUN INTEGRITY SELF-CHECK
PROGRAM ALTERED?
YES
NO
RUN PROGRAM WITH BASIC FUNCTIONALITY
OFFER EXTENDED FUNCTIONALITY
OFFER ACCEPTED?
NO
YES
TRIGGER COUNTERMEASURES
EXIT
REQUEST USAGE AUTHORIZATION
MSG
A
(TO FIG. 4)
AUTHORIZATION RECEIVED?
NO
MSG
B
(FROM FIG. 4)
YES
AUTHORIZATION VALID?
NO
YES
RUN INTEGRITY SELF-CHECK
PROGRAM ALTERED?
YES
NO
TRIGGER COUNTERMEASURES
EXIT
RUN PROGRAM WITH EXTENDED FUNCTIONALITY
REPORT USAGE
MSGS
C
(TO FIG. 4)
AUTHORIZATION STALE?
NO
YES

FIG. 3

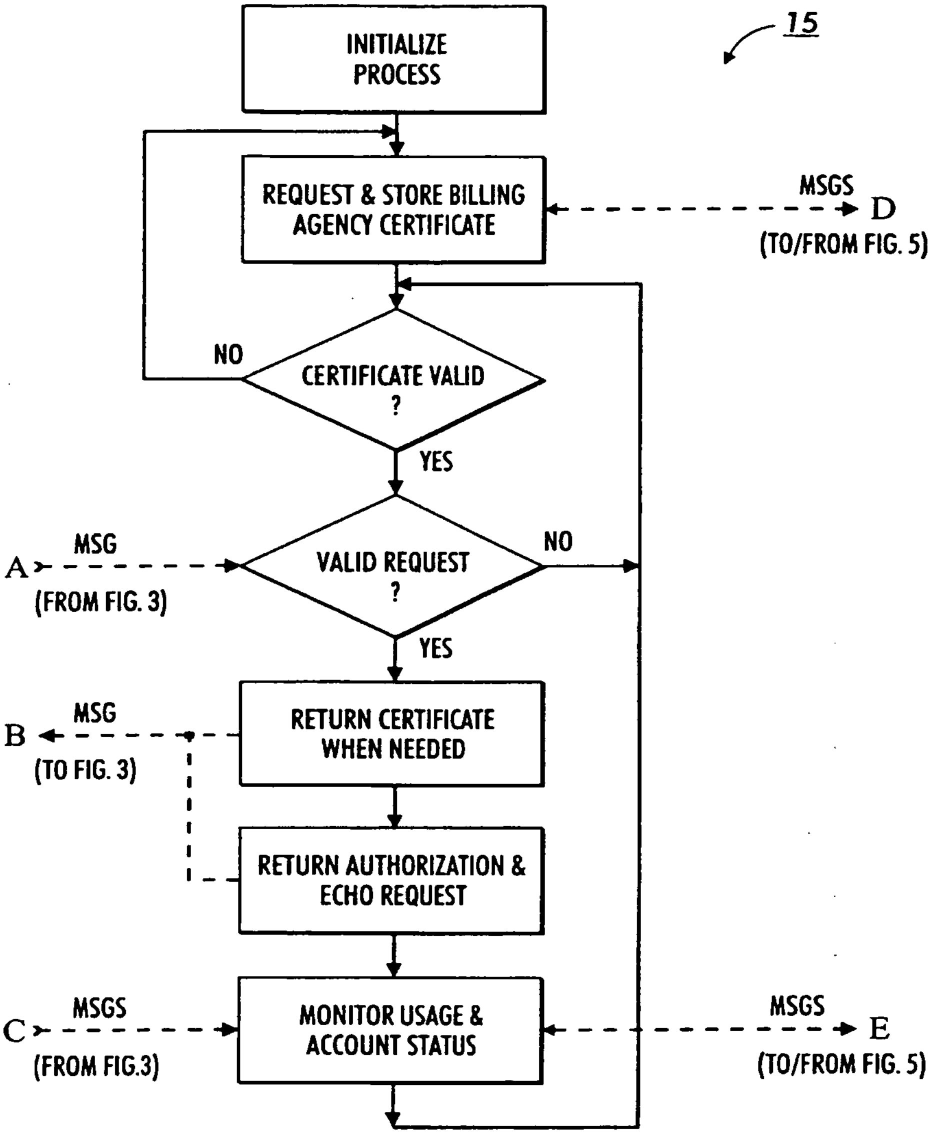

15
INITIALIZE PROCESS
REQUEST & STORE BILLING AGENCY CERTIFICATE
MSGS
D
(TO/FROM FIG. 5)
NO
CERTIFICATE VALID ?
YES
A
MSG
(FROM FIG. 3)
VALID REQUEST ?
NO
YES
MSG
B
(TO FIG. 3)
RETURN CERTIFICATE WHEN NEEDED
RETURN AUTHORIZATION & ECHO REQUEST
C
MSGS
(FROM FIG.3)
MONITOR USAGE & ACCOUNT STATUS
MSGS
E
(TO/FROM FIG. 5)

FIG. 4